IN THE CLAIMS

Please amend the claims as follows:

1. (Cancelled)

2. (Cancelled)

3. (Previously Presented) The method as defined in claim 21, wherein the content that is loaded into the in-memory copy comprises one or more images with textual content embedded therein including at least one of an in-line GIF image and an in-line JPEG image.

4. (Previously Presented) The method as defined in claim 21, wherein the executing a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy further comprises executing on the server one or more Java applets with textual content embedded therein.

5. (Previously Presented) The method as defined in claim 21, wherein the executing a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy further comprises executing the client-side scripting code on the server which directs the loading of web documents selected from the group of documents consisting of in-line frames, frames, and equivalents.

6. (Previously Presented) The method as defined in claim 4, wherein the executing a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy further comprises executing on the server one or more Java Script components with textual content embedded therein.

7. (Previously Presented) The method as defined in claim 21, wherein the retrieving the dynamic data document further comprises performing the following sub-steps of:

initializing a first list with seed values;

checking if there are any URLs to be processed and in response that any URL exists to be processed then performing the following sub-steps of:

determining if a URL is in a second list; and in response that a URL is not in the second list then performing the following sub-steps of:

inserting the URL into the first list;

scheduling the URL for crawling;

crawling the URL when scheduled to do so;

removing the URL from the first list after the scheduled crawling;

entering the URL into the second list; and

repeating the checking step until there are no more URLs to be processed;

where if the determining step determines that the URL is in the second list then repeating the checking step until there are no more URLs to be processed.

8. (Original) The method as defined in claim 7, wherein the sub-step of initializing a first list with seed values further includes the list being a URL pool.

9. (Original) The method as defined in claim 7, wherein the sub-step of determining if a URL is in a second list further includes the second list being a visited pool.

10. (Previously Presented) The method as defined in claim 7, wherein the sub-step of crawling further comprises the sub-steps of:

issuing an HTTP command to a web server named in the URL;

receiving contents of an HTML page as a result of the issued HTTP command; and

passing on the contents of the HTML page to a Page Rendering subroutine.

11. (Previously Presented) The method as defined in claim 10, further including the sub-steps performed by the Page Rendering subroutine comprising:

receiving the contents of the HTML page in the Page Rendering subroutine;

building an in-memory copy of a web-browser layout for the HTML page and if more data is needed to properly form the copy, then performing the sub-steps of:

requesting additional web-based information;

gathering this additional web-based information;

inserting any URLs associated with this additional web-based information into the second list and a URL cache;

building a final amended in-memory copy; and

forwarding the final amended in-memory copy to an Extraction subroutine;

wherein, if no more data is needed to properly form the in-memory representation, then forwarding the in-memory copy to the Extraction subroutine.

12. (Original) The method as defined in claim 11, further including the sub-steps performed by the Page Extraction subroutine comprising:

accessing a set of memory structures of the Page Renderer;

copying a text portion of the structures into a text map;

inspecting any in-line GIF and JPEG image references in the memory structures;

extracting alternate text attributes;

adding the alternate text attributes to a text map;

invoking an optical character recognition engine;

analyzing any in-line GIF and JPEG images using the optical character recognition engine for text content;

extracting text content from the GIF and JPEG images;

adding text content from the images to the text map; and

forwarding the text map to a Page Summarizer subroutine.

13. (Original) The method as defined in claim 12, further including the sub-steps performed by the Page Summarizer subroutine comprising:

receiving a text map from the Page Extractor subroutine;

processing the text map in an application-specific manner;

applying data extraction patterns to the text map;

translating resultant data from the applying step;

forwarding any URLs present in the text map to a manager subroutine; and

forwarding any extracted data and metadata to application logic.

Claims 14-20 (Cancelled)

21. (Previously Presented) A method for indexing dynamic data documents, the method comprising:

retrieving, to a server, with a web crawler from a network address, a dynamic data document with client-side scripting code therein;

executing, at the server, a web-browser, as part of the web crawler, wherein the web-browser renders an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic data document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document;

executing, at the server instead of a client system, a browser scripting engine as part of the web-browser, wherein the browser scripting engine executes the client-side scripting code and loads content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the dynamic data document is rendered at a user's web-browser and viewed by a user in the user's web-browser running on the client system when all the dynamic data is

viewed; and

indexing, at the server, the content in the memory, wherein the content being indexed is the content which has been loaded by the browser scripting engine in order to index the dynamic data document as if being viewed by the user in the user's web-browser on the client system.

22. (Previously Presented) The method of claim 21, wherein the indexing further comprises:

analyzing and summarizing, at the server, the final web-browser display representation of the dynamic data document to produce a text map for the dynamic data document; and

using optical character recognition on the content that has been loaded into the in-memory copy to extract textual content for adding to the textual map for the dynamic data document.

Claims 23-25 (Cancelled)